

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-119986  
(43)Date of publication of application : 30.04.1999

(51)Int.CI. G06F 9/06  
G06F 13/00  
G06F 15/16

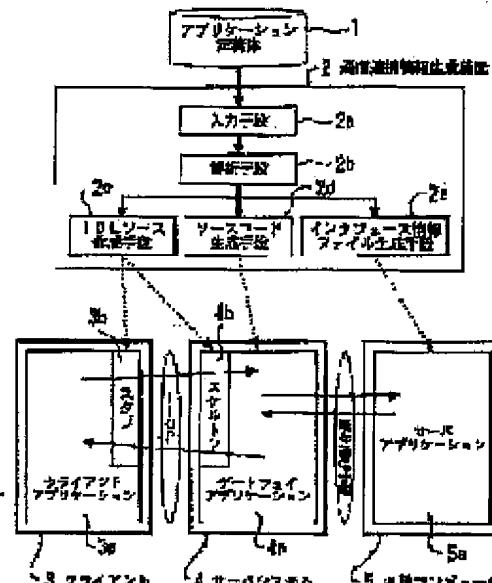
(21)Application number : 09-288643 (71)Applicant : FUJITSU LTD  
(22)Date of filing : 20.10.1997 (72)Inventor : UEDA TAKUSHI

## (54) COMMUNICATION LINK INFORMATION GENERATOR, THREE-HIERARCHY CLIENT/SERVER SYSTEM AND RECORD MEDIUM RECORDED WITH COMMUNICATION LINK INFORMATION GENERATION PROGRAM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To generate the link information of communication at the time of providing a three-hierarchy system by applying the general-purpose computer of a non-distributed object environment to a distributed object environment.

**SOLUTION:** For a communication link information generator 2, an application definition 1 for defining the information of communication among three parties into one is read by an input means 2a and analyzed by an analysis means 2b. An interface definition language(IDL) source preparing means 2c prepares an IDL source for a client/server system from that application definition 1, the source code of the application of a server system located as a gateway is generated by a source code generating means 2d, and an interface information file for the server application of the general-purpose computer is generated by an interface information file generating means 2e, so that linking of different environments is realized.



### LEGAL STATUS

[Date of request for examination] 21.09.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3597356

[Date of registration] 17.09.2004

[Number of appeal against examiner's decision of rejection]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-119986

(13)公開日 平成11年(1999)4月30日

(51)Int.Cl.<sup>a</sup>  
G 0 6 F 9/06  
13/00  
15/16

識別記号  
5 3 0  
3 5 7  
3 7 0

F I  
G 0 6 F 9/06  
13/00  
15/16

5 3 0 V  
3 5 7 Z  
3 7 0 N

審査請求 未請求 請求項の数9 O.L (全10頁)

(21)出願番号 特願平9-286643

(22)出願日 平成9年(1997)10月20日

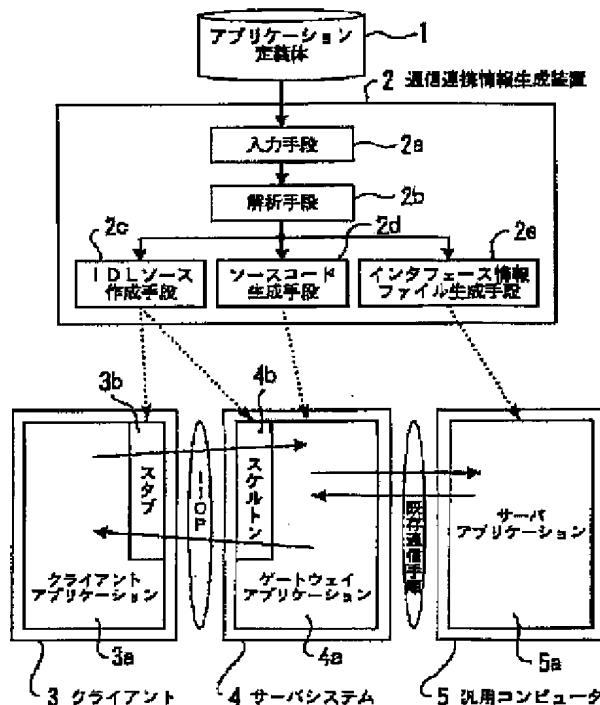
(71)出願人 000005223  
富士通株式会社  
神奈川県川崎市中原区上小田中4丁目1番  
1号  
(72)発明者 上田 琢司  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内  
(74)代理人 弁理士 服部 鋼巖

(54)【発明の名称】 通信連携情報生成装置、3階層クライアント/サーバシステムおよび通信連携情報生成プログラムを記録した媒体

(57)【要約】

【課題】 通信連携情報生成装置に関し、非分散オブジェクト環境の汎用コンピュータを分散オブジェクト環境に適用して3階層システムにするとときの通信の連携情報を生成することを目的とする。

【解決手段】 通信連携情報生成装置2は3者間の通信情報を一つに定義したアプリケーション定義体1を入力手段2aが読み込み、解析手段2bで解析する。そのアプリケーション定義体1からインターフェース定義言語(1DL)ソース作成手段2cがクライアント/サーバシステムのためのインターフェース定義言語ソースを作成し、ソースコード生成手段2dがゲートウェイと位置付けられたサーバシステムのアプリケーションのソースコードを生成し、インターフェース情報ファイル生成手段2eが汎用コンピュータのサーバアプリケーションのためのインターフェース情報ファイルを生成することで、異なる環境の連携を実現させる。



## 【特許請求の範囲】

【請求項1】 クライアントアプリケーションとゲートウェイアプリケーションと汎用コンピュータ上のサーバアプリケーションとで構成される3階層クライアント/サーバシステムの通信連携情報を生成する通信連携情報生成装置において、

各アプリケーションのインターフェース情報を定義したアプリケーション定義体を読み込む入力手段と、

読み込んだ前記アプリケーション定義体を解析する解析手段と、

前記アプリケーション定義体からクライアントアプリケーションとゲートウェイアプリケーションとの通信のためのインターフェース情報を定義したインターフェース定義言語ソースを作成するインターフェース定義言語ソース作成手段と、

前記アプリケーション定義体から汎用コンピュータへの通信情報を含んだゲートウェイアプリケーション自体のソースコードを生成するソースコード生成手段と、

前記アプリケーション定義体から前記サーバアプリケーションがゲートウェイアプリケーションと通信処理を行うためのインターフェース情報ファイルを生成するインターフェース情報ファイル生成手段と、

を備えていることを特徴とする通信連携情報生成装置。

【請求項2】 前記アプリケーション定義体は、オブジェクト情報と、ネットワーク上の汎用コンピュータを識別するための通信あて先と、クライアントアプリケーションとゲートウェイアプリケーションとサーバアプリケーションとの間のインターフェース情報を含んでいることを特徴とする請求項1記載の通信連携情報生成装置。

【請求項3】 前記インターフェース定義言語ソース作成手段により作成されるインターフェース定義言語ソースは、これをコンパイルすることによってクライアントアプリケーションに組み込まれるスタブおよびゲートウェイアプリケーションに組み込まれるスケルトンが生成されるソースであることを特徴とする請求項1記載の通信連携情報生成装置。

【請求項4】 前記ソースコード生成手段により生成されるソースコードは、前記クライアントアプリケーションのオブジェクトから呼び出されたときに指定された汎用コンピュータにパススルーの処理をするオブジェクトのソースコードであることを特徴とする請求項1記載の通信連携情報生成装置。

【請求項5】 前記インターフェース情報ファイル生成手段により生成されるインターフェース情報ファイルは、前記サーバアプリケーションのプログラムのコンパイル時に前記サーバアプリケーションに組み込まれるライブラリ関数のファイルであることを特徴とする請求項1記載の通信連携情報生成装置。

【請求項6】 クライアントおよびサーバシステムに汎

用コンピュータを接続して構成される3階層クライアント/サーバシステムにおいて、

一つのアプリケーション定義体から生成したインターフェース定義言語ソースをコンパイルすることによって作られたインターフェース情報を組み込まれたクライアントアプリケーションが配置されているクライアントと、前記クライアントアプリケーションとは分散オブジェクト指向技術を利用して通信するよう前記アプリケーション定義体から生成したインターフェース定義言語ソースをコンパイルすることによって作られたインターフェース情報を組み込まれかつ前記アプリケーション定義体から生成した汎用コンピュータとの通信連携情報を組み込まれたゲートウェイアプリケーションが配置されているサーバシステムと、

前記ゲートウェイアプリケーションとは既存通信手段により通信するよう前記アプリケーション定義体から生成したインターフェース情報を組み込んだサーバアプリケーションが配置されている汎用コンピュータと、を備えていることを特徴とする3階層クライアント/サーバシステム。

【請求項7】 前記クライアントアプリケーションおよび前記ゲートウェイアプリケーションのオブジェクトは、分散オブジェクト環境内の位階透過性がネーミングサービスによって保持されていることを特徴とする請求項6記載の3階層クライアント/サーバシステム。

【請求項8】 前記ゲートウェイアプリケーションの汎用コンピュータとの通信連携情報は、前記サーバシステムと汎用コンピュータとを接続しているネットワーク上の汎用コンピュータを…意に識別する通信あて先を含んでいることを特徴とする請求項6記載の3階層クライアント/サーバシステム。

【請求項9】 クライアントアプリケーションとゲートウェイアプリケーションと汎用コンピュータ上のサーバアプリケーションとで構成される3階層クライアント/サーバシステムの通信連携情報を生成するプログラムを記録したコンピュータ読み取り可能な媒体において、各アプリケーションのインターフェース情報をおよび前記ゲートウェイアプリケーションと前記サーバアプリケーションとの間の通信情報を定義されたアプリケーション定義体を解析する手段、クライアントアプリケーションとゲートウェイアプリケーションとの間の通信のためのインターフェースを定義したインターフェース定義言語ソースを作成する手段、ゲートウェイアプリケーション自体のソースコードを生成する手段、およびサーバアプリケーションがゲートウェイアプリケーションと通信処理を行うためのインターフェース情報ファイルを生成する手段としてコンピュータを機能させるためのプログラムを記録した媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は通信連携情報生成装置に関する、特に分散オブジェクト指向技術を利用して通信処理を行うクライアント/サーバシステムと分散オブジェクト環境を持たない汎用コンピュータとで構成されるような3階層クライアント/サーバシステムにおけるアプリケーション間の通信連携情報を生成する通信連携情報生成装置に関する。

【0002】クライアント/サーバシステムはアプリケーション全体の機能をクライアントとサーバとに分散配置し、ネットワークを通じて全体で連携処理するものである。このような形態のシステムでは、クライアントはユーザインターフェースを主目的とし、サーバはクライアントにてユーザとの対話処理の過程にて発生した要求を受けて処理することを主目的としている。クライアントはユーザインターフェースに付随して起きるイベントに応答して処理する機能を備えているため、機能が増えてくれば、必然的にクライアントの負担が増える傾向にある。そこで、機能の密な結合を回避し、さらにシステムに拡張性および柔軟性を持たせるために、システムを階層化することが提案されており、クライアント層とサーバ層との間に少なくとも一つの階層を介在させて多階層システムの形態を探るようになってきている。

【0003】一方、クライアント/サーバシステムについて、それらのアプリケーション間の通信においても、信頼性、拡張性、柔軟性などが求められており、そのような要求に応えるものとして分散オブジェクト環境がある。

【0004】オブジェクト指向開発は、アプリケーションをオブジェクトと見ることでプログラム開発効率を高める技術として浸透しつつある。このうち、CORBA (Common Object Request Broker Architecture) は、分散したシステム上のサーバアプリケーションを、クライアントアプリケーションが位置や実装を認識することなく呼び出し可能とする技術として、OMG (Object Management Group) 団体が定めた分散オブジェクト指向技術のための規約である。

【0005】クライアント/サーバシステムにおいて、サーバアプリケーションのインターフェース情報を、CORBAによってネットワーク上に公開して共用することにより、クライアントアプリケーションは、サーバアプリケーションをあたかもローカルシステム内のプログラム呼び出しのように容易に利用することができる。そして、オブジェクト同士で通信を行うための機能およびソフトウェアはORB (object request broker) と呼ばれ、ORB同士を接続する通信手順としてIIOP (Internet Inter-ORB Protocol) が採用されている。

【0006】

【従来の技術】図7は3階層システムの構築例を示す図

である。この図において、3階層システムは、クライアント100と、サーバシステム110と、データサーバ120とから構成されている。クライアント100には、クライアントアプリケーション101が配置され、サーバシステム110にはサーバアプリケーション111が配置され、データサーバ120にはサーバアプリケーション121が配置されている。したがって、サーバアプリケーション111はサーバアプリケーション121のクライアントアプリケーションとしての機能も有する。ここで、クライアントアプリケーション101のオブジェクトAがサーバアプリケーション111のオブジェクトBを呼び出し、そのオブジェクトBがサーバアプリケーション121のオブジェクトCを呼び出すという例を示している。これらオブジェクト間のメッセージ交換はインターフェース情報ファイルと、ORB機構と、ORB間通信プロトコルによって実現される。

【0007】インターフェース情報ファイルは、オブジェクトに関するインターフェース情報をCORBA規約で定められた文法に従ったインターフェース定義言語 (IDL) で記述し、それを専用のIDLコンパイラでコンパイルすることにより、クライアント、サーバそれぞれのアプリケーションに対して作成される。CORBAでは、このインターフェース情報ファイルを、クライアント側はスタブ、サーバ側はスケルトンと呼んでいる。

【0008】クライアントアプリケーション101にはスタブ102およびORB機構103があり、サーバアプリケーション111にはORB機構112、スケルトン113、スタブ114およびORB機構115があり、サーバアプリケーション121にはORB機構122およびスケルトン123がある。ここで、スタブ102およびスケルトン113はオブジェクトBに関する情報を記述したIDLファイル131をコンパイルすることによって生成されたインターフェース情報ファイルであり、スタブ114およびスケルトン123はオブジェクトCに関する情報を記述したIDLファイル132をコンパイルすることによって生成されたインターフェース情報ファイルである。

【0009】また、ORB機構103とORB機構112との間、およびORB機構115とORB機構122との間の通信はCORBAの標準プロトコルであるIIOPが使用される。

【0010】さらに、ネットワーク上には別のサーバ140があり、これにはネーミングサービス141が搭載されている。このネーミングサービス141は、アプリケーションのオブジェクトが必要としているオブジェクトがネットワーク上のどこにあるかの問い合わせに応えるためのもので、オブジェクトが位置するサーバのアドレスとともに、オブジェクトを名前で管理しているデータベースである。これにより、クライアントアプリケーションはサーバアプリケーションのオブジェクトを名前

で呼び出すことにより、ORB機構はネーミングサービス141を参照し、ここで得られたサーバアプリケーションのオブジェクトが位置するサーバシステムのアドレスをもとにそのサーバシステムに要求を出すことによって利用可能になる。

【0011】次に、このように構築された3階層システムにおけるオブジェクト呼び出しについて説明する。まず、クライアントアプリケーション101のオブジェクトAがオブジェクトBを呼び出す。これは、スタブ102内に展開されたオブジェクトBに対応するオブジェクト（オペレーション）を呼び出すことになる。このオペレーションには、通信のためのインターフェース情報は持たないので、ORB機構103はネーミングサービス141に問い合わせて、得られたサーバシステム110のアドレスをもとにIOPプロトコルを使用してサーバシステム110のORB機構112にクライアントアプリケーション101からの要求を送る。ORB機構112を介して要求を受けたオブジェクトBはその要求をオブジェクトCに転送する。この場合も、同様にしてネーミングサービス141を利用して、ネットワーク上のデータサーバ120のアドレスを取得して通信を行う。データサーバ120のサーバアプリケーション121はオブジェクトBからの要求を処理する。オブジェクトCは処理した結果をオブジェクトBに返信する。オブジェクトBはオブジェクトCからの応答をオブジェクトAに返信する。これにより、クライアントアプリケーション101は処理した結果を受け取ることになる。

【0012】ところで、以上のような3階層システムにおいて、3階層目のデータサーバ120に大型の汎用コンピュータを使いたいという要求がある。このとき、2階層目のサーバシステム110はクライアント100からの要求を汎用コンピュータに単純に転送するだけのゲートウェイ処理となる。このような形態を探ることにより、システム全体の信頼性・性能を向上させることができ、しかも、既に運用しているデータベースシステムの再利用、有効利用が可能になる。

#### 【0013】

【発明が解決しようとする課題】CORBAの技術は標準的なオペレーティングシステムやハードウェアをもとにして作られたオープン仕様のコンピュータに適用される。一方、汎用コンピュータは独自のアーキテクチャに基づいて作られたものであるため、3階層システムの3階層目に汎用コンピュータを適用する場合に、次のような問題点が発生する。

【0014】まず、本来、分散オブジェクト指向技術を持たない汎用コンピュータ上に分散オブジェクト指向技術と連携させるためのORB機構を持つ必要がある。CORBAではオブジェクト名をネットワーク一意の名前として管理しているために、たとえ3者間のインターフェースが同じであっても、クライアントアプリケーション

とゲートウェイアプリケーションとのインターフェースを定義するIDLソースと、ゲートウェイアプリケーションと汎用コンピュータ上のサーバアプリケーションとのインターフェース情報を定義するIDLソースとの2つのIDL定義が必要となる。

【0015】2階層目のサーバシステムをクライアントからの要求を単に汎用コンピュータに転送するゲートウェイとしてシステムを構築する場合でも、開発者はクライアントからの要求を受けるスケルトンの部分と、汎用コンピュータに向かうスタブの部分と、ゲートウェイオブジェクトの部分とをコーディングしなければならず、ゲートウェイのオブジェクトと汎用コンピュータ上のサーバアプリケーションのオブジェクトとの関連付けがゲートウェイソースの作成時のインプリメントマターとなる。

【0016】そして、ORB機構は、ネーミングサービスを参照して通信先システムを決定するために、ネーミングサービスが搭載されるコンピュータとの通信を必要とする。したがって、クライアントからサーバシステムに要求を出すときにネーミングサービスを利用し、サーバシステムから汎用コンピュータに要求を出すときもまたネーミングサービスを利用することになるため、クライアント／サーバだけで構成される分散オブジェクトシステムに比べ、通信処理に時間がかかり、システムの性能が落ちるという問題点があつた。

【0017】本発明はこのような点に鑑みてなされたものであり、3階層システムに分散オブジェクト環境を持たない汎用コンピュータを適用した場合に、ゲートウェイの作成、すなわち、コーディングが不要な通信連携情報生成装置およびそのプログラムを記録した媒体と、通信時間が長くならない3階層クライアント／サーバシステムを提供することを目的とする。

#### 【0018】

【課題を解決するための手段】図1は上記目的を達成する本発明の原理図である。図1の上部には、分散オブジェクト環境にあるクライアント／サーバシステムと非分散オブジェクト環境の汎用コンピュータとを連携させる情報を生成する通信連携情報生成装置が示されている。クライアント／サーバシステムと汎用コンピュータとの3者間の通信連携情報はアプリケーション定義体1に定義されている。このアプリケーション定義体1には、オブジェクト情報と、汎用コンピュータの通信あて先と、クライアントアプリケーションとゲートウェイアプリケーションとサーバアプリケーションとの間のインターフェース情報とが記述されている。通信連携情報生成装置2は、アプリケーション定義体1を読み込む入力手段2aと、そのアプリケーション定義体1を解析する解析手段2bと、クライアント／サーバシステムのためのインターフェース定義言語（1DL）ソースを作成するインターフェース定義言語ソース作成手段2cと、サーバシステム

7

のゲートウェイアプリケーションのソースコードを生成するソースコード生成手段2 d と、汎用コンピュータのサーバアプリケーションのためのインターフェース情報ファイルを生成するインターフェース情報ファイル生成手段2 e とを備えている。

【 0019 】この通信連携情報生成装置2 によれば、入力手段2 a がアプリケーション定義体1 を読み込み、解析手段2 b がその内容を解析する。その解析結果はインターフェース定義言語ソース作成手段2 c と、ソースコード生成手段2 d と、インターフェース情報ファイル生成手段2 e とに振り分けられる。インターフェース定義言語ソース作成手段2 c はクライアントのクライアントアプリケーションに組み込まれるスタブおよびサーバシステムのゲートウェイアプリケーションに組み込まれるスケルトンのためのソースを生成する。ソースコード生成手段2 d は汎用コンピュータの通信あて先を考慮したゲートウェイアプリケーションのソースコードを出力する。インターフェース情報ファイル生成手段2 e は汎用コンピュータのサーバアプリケーションに組み込まれる、ゲートウェイアプリケーションとのインターフェース情報のファイルを出力する。これにより、通信連携情報生成装置2 がゲートウェイアプリケーションのソースコードと汎用コンピュータ用のインターフェース情報ファイルを生成するようにしたことによりゲートウェイアプリケーションのコーディングが不要となる。

【 0020 】また、本発明によれば、クライアントアプリケーションとゲートウェイアプリケーションと汎用コンピュータ上のサーバアプリケーションとで構成される3 階層クライアント／サーバシステムの通信連携情報を生成するプログラムを記録したコンピュータ読み取り可能な媒体において、各アプリケーションのインターフェース情報および前記ゲートウェイアプリケーションと前記サーバアプリケーションとの間の通信情報が定義されたアプリケーション定義体を解析する手段、クライアントアプリケーションとゲートウェイアプリケーションとの間の通信のためのインターフェースを定義したインターフェース定義言語ソースを作成する手段、ゲートウェイアプリケーション自身のソースコードを生成する手段、およびサーバアプリケーションがゲートウェイアプリケーションと通信処理を行うためのインターフェース情報ファイルを生成する手段としてコンピュータを機能させるためのプログラムを記録した媒体が提供される。

【 0021 】この媒体に記録された通信連携情報生成プログラムをコンピュータに実行させることにより、アプリケーション定義体を解析する手段と、インターフェース定義言語ソースを作成する手段と、ゲートウェイアプリケーションのソースコードを生成する手段と、インターフェース情報ファイルを生成する手段との各機能がコンピュータによって実現できる。

【 0022 】さらに、図1 の下部には、汎用コンピュー

8

タを含む3 階層クライアント／サーバシステムが示されており、クライアント3 と、サーバシステム4 と、汎用コンピュータ5 とから構成されている。クライアント3 はクライアントアプリケーション3 a を搭載し、このクライアントアプリケーション3 a には、インターフェース定義言語ソース作成手段2 c からのソースをもとに作られスタブ3 b が組み込まれている。サーバシステム4 はゲートウェイの位置付けとして動作し、分散オブジェクト指向技術を持たない大型の汎用コンピュータ5 上のアプリケーションと連携処理を行うゲートウェイアプリケーション4 a を搭載し、このゲートウェイアプリケーション4 a はソースコード生成手段2 d からのソースコードをもとに作成され、そのとき、インターフェース定義言語ソース作成手段2 c からのソースをもとに作られスケルトン4 b が組み込まれる。そして、汎用コンピュータ5 はサーバアプリケーション5 a を搭載している。ここで、クライアントアプリケーション3 a とゲートウェイアプリケーション4 a とが、分散オブジェクト指向技術(CORBA)を利用して処理を行う。ゲートウェイアプリケーション4 a と分散オブジェクト指向技術を持たない汎用コンピュータのサーバアプリケーション5 a との通信処理は既存通信手順を利用する。既存通信手順では、汎用コンピュータを特定するための情報として、通信あて先を利用する。

【 0023 】ゲートウェイアプリケーション4 a はクライアントアプリケーション3 a からの要求を汎用コンピュータ5 のサーバアプリケーション5 a へ通知し、サーバアプリケーション5 a の処理結果をクライアントアプリケーションに応答として通知する。これにより、クライアントアプリケーション3 a は、分散オブジェクト環境から、汎用コンピュータ5 上のサーバアプリケーション5 a を利用することができるようになる。このとき、クライアントアプリケーション3 a がゲートウェイアプリケーション4 a に要求を出すときは、そのインターフェース情報をネーミングサービスを利用して取得するが、ゲートウェイアプリケーション4 a がサーバアプリケーション5 a へ要求を転送するときは既存通信手順を利用する。既存通信手順では、汎用コンピュータの通信あて先を直接利用するので、すぐに、目的のコンピュータとの通信処理を開始することができる。これにより、ゲートウェイアプリケーション4 a とサーバアプリケーション5 a との間の通信処理については、ネーミングサービスを搭載したコンピュータとの通信が必要なくなるので処理時間がORB機構に比べ半減するのとともに、汎用コンピュータ上にORB機構を作成する必要がなくなる。

【 0024 】

【 発明の実施の形態】まず、本発明の概略について図面を参照して説明する。図1 は本発明の通信連携情報生成装置および3 階層クライアント／サーバシステムの構成

を示す図である。図1において、その上部には、分散オブジェクト環境にあるクライアント/サーバシステムと非分散オブジェクト環境の汎用コンピュータとを連携させる情報を生成する通信連携情報生成装置が示されている。クライアント/サーバシステムと汎用コンピュータとの3者間の通信連携情報は、一つのアプリケーション定義体1に定義されている。このアプリケーション定義体1には、オブジェクト情報と、汎用コンピュータをネットワーク上で識別することができる通信連携情報である通信あて先と、クライアントアプリケーションとゲートウェイアプリケーションとサーバアプリケーションとの間のインターフェース情報とが記述されている。通信連携情報生成装置2は、アプリケーション定義体1を読み込む入力手段2aと、そのアプリケーション定義体1を解析する解析手段2bと、アプリケーション定義体1からクライアント/サーバシステムのためのインターフェース定義言語(IDL)ソースを作成するインターフェース定義言語ソース作成手段2cと、アプリケーション定義体1からサーバシステムのゲートウェイアプリケーションのソースコードを生成するソースコード生成手段2dと、アプリケーション定義体1から汎用コンピュータのサーバアプリケーションのためのインターフェース情報ファイルを生成するインターフェース情報ファイル生成手段2eとを備えている。

【0025】この通信連携情報生成装置2によれば、入力手段2aがアプリケーション定義体1を読み込み、解析手段2bがその内容を解析する。その解析結果はインターフェース定義言語ソース作成手段2cと、ソースコード生成手段2dと、インターフェース情報ファイル生成手段2eとに振り分けられる。インターフェース定義言語ソース作成手段2cはクライアントのクライアントアプリケーションに組み込まれるスタブおよびサーバシステムのゲートウェイアプリケーションに組み込まれるスケルトンのためのソースを生成する。ソースコード生成手段2dは汎用コンピュータの通信あて先を考慮したゲートウェイアプリケーションのソースコードを出力する。インターフェース情報ファイル生成手段2eは汎用コンピュータのサーバアプリケーションに組み込まれる、ゲートウェイアプリケーションとのインターフェース情報のファイルを出力する。これにより、通信連携情報生成装置2がゲートウェイアプリケーションのソースコードと汎用コンピュータ用のインターフェース情報ファイルを生成するようにしたことによりゲートウェイアプリケーションのコーディングが不要となる。

【0026】また、図1の下部には、汎用コンピュータを含む3階層クライアント/サーバシステムが示されているが、これは、クライアント3と、サーバシステム4と、汎用コンピュータ5とから構成されている。クライアント3はクライアントアプリケーション3aを搭載し、このクライアントアプリケーション3aには、イン

タフェース定義言語ソース作成手段2cの出力したソースをもとに作られスタブ3bが組み込まれている。サーバシステム4はゲートウェイの位置付けとして動作し、分散オブジェクト指向技術を持たない大型の汎用コンピュータ5上のアプリケーションと連携処理を行なうゲートウェイアプリケーション4aを搭載し、このゲートウェイアプリケーション4aはソースコード生成手段2dからのソースコードをもとに作成される。そのとき、インタフェース定義言語ソース作成手段2cの出力したソースをもとに作られスケルトン4bがゲートウェイアプリケーション4aに組み込まれる。そして、汎用コンピュータ5はサーバアプリケーション5aを搭載している。【0027】ここで、クライアントアプリケーション3aとゲートウェイアプリケーション4aとが、分散オブジェクト指向技術(CORBA)を利用して通信処理を行う。一方、ゲートウェイアプリケーション4aと分散オブジェクト指向技術を持たない汎用コンピュータのサーバアプリケーション5aとの通信処理は既存通信手順を利用する。既存通信手順では、汎用コンピュータを特定するための情報として、通信あて先を利用する。【0028】ゲートウェイアプリケーション4aはクライアントアプリケーション3aからの要求を汎用コンピュータ5のサーバアプリケーション5aへ通知し、サーバアプリケーション5aの処理結果をクライアントアプリケーションに応答として通知する。これにより、クライアントアプリケーション3aは、分散オブジェクト環境から、汎用コンピュータ5上のサーバアプリケーション5aを利用することができるようになる。このとき、クライアントアプリケーション3aがゲートウェイアプリケーション4aに要求を出すときは、そのインターフェース情報をネーミングサービスを利用して取得するが、ゲートウェイアプリケーション4aがサーバアプリケーション5aへ要求を転送するときは既存通信手順を利用する。既存通信手順では、汎用コンピュータの通信あて先を直接利用するので、すぐに、目的のコンピュータとの通信処理を開始することができる。これにより、ゲートウェイアプリケーション4aとサーバアプリケーション5aとの間の通信処理については、ネーミングサービスを搭載したコンピュータとの通信が必要なくなるので処理時間がORB機構に比べ半減するのとともに、汎用コンピュータ上にORB機構を作成する必要がなくなる。

【0029】次に、本発明の実施の形態について説明するが、まず、分散オブジェクト環境にあるクライアント/サーバシステムと非分散オブジェクト環境の汎用コンピュータとで3階層システムを構成するときの3者間の通信連携情報の生成手順について説明する。

【0030】図2は3階層クライアント/サーバシステムの通信情報連携処理を説明した図である。クライアントとゲートウェイとして機能させるサーバシステムと汎

用コンピュータとの3者間の通信連携情報は、アプリケーション定義体1 1と呼ぶ一つの定義体に定義される。このアプリケーション定義体1 1では、クライアントアプリケーションとゲートウェイ、さらにサーバアプリケーションとの関連付けを行う必要があるが、CORBAのI DL定義上では、クライアント対サーバという1対1のインタフェース情報の共有とI IOPプロトコルの使用とを前提に定義するものであるため、汎用コンピュータとの通信のための通信アドレスを記述することはできない。したがって、3者間のインタフェース情報の定義を、I DL定義を一階層上げたアプリケーション定義体1 1によって定義することにしている。このとき、アプリケーション定義体1 1では、クライアントアプリケーション・ゲートウェイ・サーバアプリケーション間のインタフェース情報と、ゲートウェイとサーバアプリケーションとの間の通信情報をたとえれば次のように定義する。

【0031】図3はアプリケーション定義の一例を示す図である。アプリケーション定義体1 1では、アプリケーション定義はほぼI DL記述に似た記述になっていて、オブジェクト(オペレーション)の情報と、通信あて先と、インタフェース情報との繰り返しによって記述される。ここで、通信あて先はゲートウェイから見た汎用コンピュータ上の通信あて先であり、その通信あて先には、ネットワーク上の各汎用コンピュータを識別するホスト名や、汎用コンピュータ上のアプリケーション名が指定される。

【0032】図2に戻って、インタフェース情報および通信あて先が以上のように定義されたアプリケーション定義体1 1は定義体コンパイラ1 2に入力される。定義体コンパイラ1 2では、アプリケーション定義体1 1を解析し、クライアントアプリケーションとゲートウェイアプリケーションのためのI DLソースファイル1 3を作成し、汎用コンピュータ上のサーバアプリケーションがゲートウェイアプリケーションと通信するためのインタフェース情報を含むインタフェース情報ファイル1 4と、ゲートウェイアプリケーションのソースコードファイル1 5を生成する。

【0033】I DLソースファイル1 3では、CORBA規約で定められた文法に従ったインタフェース情報の定義がなされているので、I DLのコンパイルにはORB準拠の標準的なI DLコンパイラ1 6が使用される。I DLコンパイラ1 6からは、クライアントアプリケーションに組み込まれるべきインタフェース情報ファイル(スタブ)1 7と、ゲートウェイアプリケーションに組み込まれるべきインタフェース情報ファイル(スケルトン)1 8とが生成される。これらインタフェース情報ファイル1 7, 1 8は、たとえばC言語ではヘッダファイルに相当するもので、アプリケーションのソースをコンパイルするときにそのアプリケーションに組み込まれ

る。インターフェース情報ファイル1 4は、CORBAでのスケルトンファイルに相当するもので、サーバアプリケーションのソースをコンパイルするときにそのサーバアプリケーションに組み込まれる。

【0034】ゲートウェイアプリケーションのためのソースコードファイル1 5は、アプリケーション定義体1 1から抽出された、汎用コンピュータの通信アドレスを含んでおり、このソースをコンパイルしてロードモジュールを作成することにより、自動的に通信処理が組み込まれる。ゲートウェイアプリケーションがパスループ処理しか行わない場合は、アプリケーション作成者はこのゲートウェイソースを追加インプリメントする必要はない。

【0035】このような定義により、クライアントアプリケーションおよびゲートウェイアプリケーションの各オブジェクトと、汎用コンピュータおよびその上のサーバアプリケーションとが一意に関係付けができる。また、上記のような定義によると、分散オブジェクト環境下にあるオブジェクトをネットワーク上の複数の汎用コンピュータ各々と関係付けることができる。

【0036】図4は3階層クライアント/サーバシステムの連携処理のためのフローチャートである。まず、アプリケーション定義体1 1にインタフェース情報と汎用コンピュータの通信アドレスを定義する(ステップS 1)。次に、定義体コンパイラ1 2でアプリケーション定義体1 1をコンパイルし、I DLソースファイル1 3、インタフェース情報ファイル1 4、ソースコードファイル1 5を生成する(ステップS 2)。

【0037】ここで、I DLソースファイル1 3については、I DLコンパイラ1 6でコンパイルされ(ステップS 3)、これから生成されたインタフェース情報ファイル1 7、すなわち、スタブファイルを利用したクライアントアプリケーションを作成する(ステップS 4)。また、ステップS 2で生成されたソースコードファイル1 5と、ステップS 3でのコンパイルの結果、生成されたインタフェース情報ファイル1 8、すなわち、スケルトンファイルを使用してゲートウェイアプリケーションのオブジェクトを作成する(ステップS 5)。さらに、ステップS 2で生成されたインタフェース情報ファイル1 4を利用してサーバアプリケーションを作成する(ステップS 6)。そして、各アプリケーションをクライアント、サーバ、汎用コンピュータに搭載することで3者間の連携処理を行うことになる(ステップS 7)。

【0038】図5は連携処理された3階層クライアント/サーバシステムの構成を示す説明図である。図5において、クライアントアプリケーション2 0があり、このクライアントアプリケーション2 0にはI DLコンパイラ1 6でコンパイルされたインタフェース情報ファイル1 7が組み込まれている。また、ゲートウェイアプリケーション2 1は、I DLコンパイラ1 6でコンパイルさ

れたインターフェース情報ファイル18と、オブジェクト22と、サーバアプリケーション24との連携処理を行う連携処理部23とで構成される。サーバアプリケーション24は連携処理部25と、定義体コンパイラ12によって生成されたインターフェース情報ファイル14と、たとえばデータベース処理を含むサーバ処理の部分からなる、たとえばCOBOL(common business oriented language)のアプリケーションである。インターフェース情報ファイル14はたとえばCOBOLでのライブラリ関数ファイルである。

【0039】クライアントアプリケーション20はクライアント処理、すなわち、サーバアプリケーション24の呼び出しと処理結果の受信を行う。ゲートウェイアプリケーション21のオブジェクト22は汎用コンピュータのサーバアプリケーション24との間で既存通信手順を利用するための通信あて先の情報(たとえば、通信あて先=S1)を有しており、ゲートウェイ処理を行う。サーバアプリケーション24はたとえば通信あて先=S1を有し、オブジェクト22の転送先と同じ通信アドレスを有しているものとする。

【0040】次に、この3階層クライアント/サーバシステムの動作について説明する。図6は3階層クライアント/サーバシステムの動作の流れを示すフローチャートである。まず、クライアントアプリケーション20がインターフェース情報ファイル(スタブ)17を利用してゲートウェイアプリケーション21のゲートウェイオブジェクト22を呼び出す(ステップS11)。ゲートウェイオブジェクト22はクライアントアプリケーション20からの要求内容を既存通信手順に変換し、既存通信手順を使ってサーバアプリケーション24に送信する(ステップS12)。サーバアプリケーション24が既存通信手順によって受けたクライアントアプリケーション20からの要求内容に従ってデータベースを更新し、既存通信手順でゲートウェイオブジェクト22に返信する(ステップS13)。ゲートウェイオブジェクト22は既存通信手順によって受けた返信内容をインターフェース情報ファイル(スケルトン)18を利用してクライアントアプリケーション20に返信する(ステップS14)。クライアントアプリケーション20はその返信内容によってデータベースの更新結果を受信する(ステップS15)。

【0041】

【発明の効果】以上説明したように本発明では、一つのアプリケーション定義体より3階層システム間の通信連携情報を得るよう構成した。このため、分散オブジェクト環境(ORB機構)を持たない汎用コンピュータ上のアプリケーションと分散オブジェクト環境上のクライアントアプリケーションとの通信連携処理を行うために必要であったサーバアプリケーションと汎用コンピュータ

上のアプリケーションとの通信処理およびインターフェース情報のインプリメントの必要がない。ゲートウェイと汎用コンピュータとの間の通信処理を既存通信手順を利用したことによりネーミングサービスを利用しない分、通信処理時間が半減され、クライアントアプリケーションに対し処理性能を向上することができる。

【0042】クライアントアプリケーションは、分散オブジェクト環境外である汎用コンピュータ上のアプリケーションとの連携を意識する必要がなくなり、あたかも汎用コンピュータ上のアプリケーションの処理を、ローカルシステムでのプログラム呼び出しのイメージで行うことができる。

【0043】クライアントアプリケーション・サーバアプリケーション間のインターフェースはOMG準拠のIDL記述であるため、そのソースファイルのコンパイルに一般的なDLコンパイラを利用することができます、クライアント側の標準性を保つことができる。

【0044】インターフェース情報が保証されたゲートウェイのソースコードが生成されるため、ゲートウェイのコーディングが不要であり、アプリケーション開発者の作業負担を軽減することができる。

#### 【図面の簡単な説明】

【図1】本発明の通信連携情報生成装置および3階層クライアント/サーバシステムの構成を示す図である。

【図2】3階層クライアント/サーバシステムの通信情報連携処理を説明した図である。

【図3】アプリケーション定義の一例を示す図である。

【図4】3階層クライアント/サーバシステムの連携処理のためのフローチャートである。

【図5】連携処理された3階層クライアント/サーバシステムの構成を示す説明図である。

【図6】3階層クライアント/サーバシステムの動作の流れを示すフローチャートである。

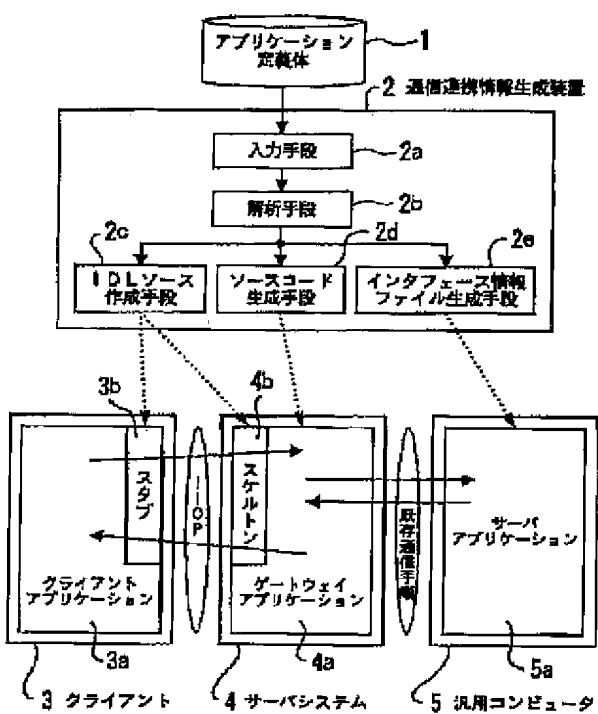
【図7】3階層システムの構築例を示す図である。

#### 【符号の説明】

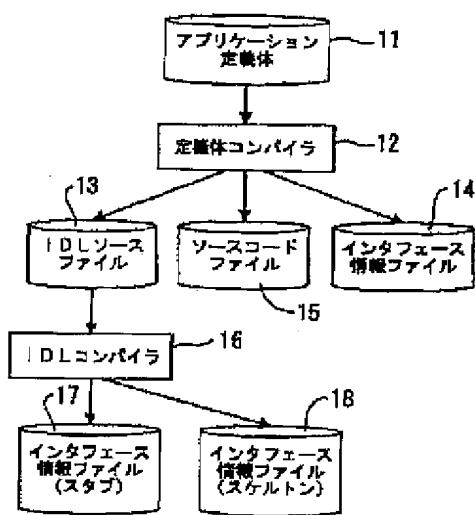
- 1 アプリケーション定義体
- 2 通信連携情報生成装置
- 2a 入力手段
- 2b 解析手段
- 2c インタフェース定義言語(IDL)ソース作成手段
- 2d ソースコード生成手段
- 2e インタフェース情報ファイル生成手段
- 3 クライアント
- 3a クライアントアプリケーション
- 3b スタブ
- 4 サーバシステム
- 4a ゲートウェイアプリケーション
- 4b スケルトン
- 5 汎用コンピュータ

## 5a サーバアプリケーション

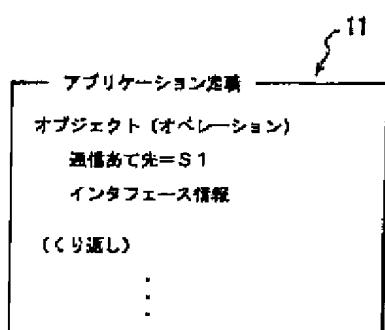
【図1】



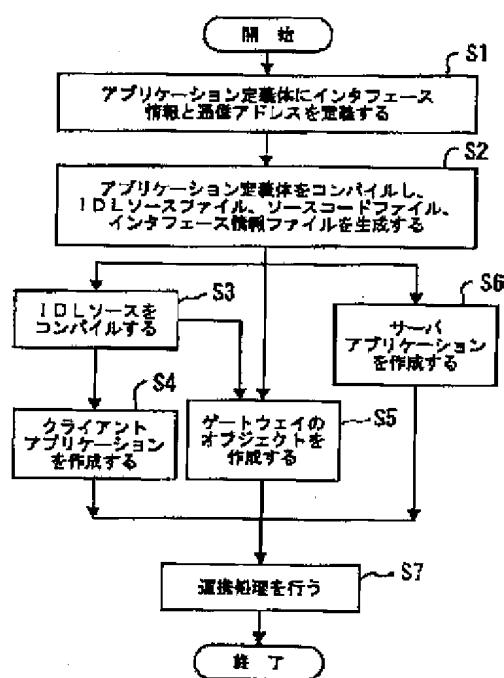
【図2】



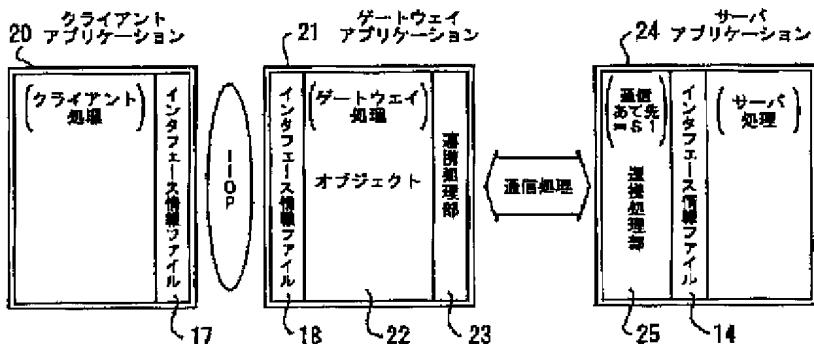
【図3】



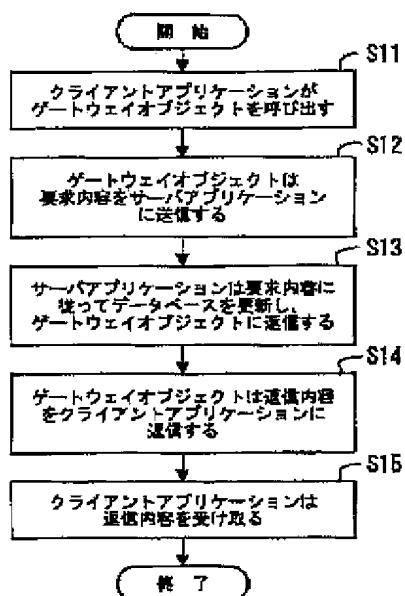
【図4】



【 図5 】



【 図6 】



【 図7 】

